Extending semantic browser style specifications for Semantic Web inferencing

Mellema, P (Pascal)

Award date:
2020

Link to publication

Extending semantic browser style specifications for Semantic Web inferencing

Degree program:	Open University of the Netherlands, Faculty of Management, Science & Technology Business Process Management & IT master's program
Course:	IM0602 BPMIT Graduation Assignment Preparation IM9806 Business Process Management and IT Graduation Assignment
Student:	P. Mellema
Identification number:	
Date:	02-02-2020
Thesis supervisor	dr. L. Rutledge
Second reader	prof. dr. ir. S. Joosten
Version number:	1.00
Status:	Final

Acknowledgement

During the past two years I have been working towards the Master's degree in Business Process Management & IT at the Open University. Sometimes it was easy, sometimes it was a challenge. What started as a hobby became a serious interest. First of all, I would like to extend my gratitude to all those people who helped and supported me in completing this thesis.

Especially I would like to thank my thesis supervisor Lloyd Rutledge, for the lessons, guidance and advices he bestowed upon me. He inspired me to work efficiently on this thesis. In addition, I would like to thank Stef Joosten for reading and providing feedback.

I would also want to express my gratefulness to my beloved wife who gave all of her time to make sure I would finish this thesis. In addition, I express my gratitude to my fantastic children who encouraged me to keep going even though they would rather I joined them in a boardgame or other family activity.

My special thanks to all friends for supporting me and keeping me on track. Last but not the least, I would like to extend my gratitude to Cor for his unconditional support during office hours.

Without the help and support of the people mentioned above, I would face many more difficulties while doing this thesis. Again, I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this thesis.

At the end! Thanks to you, reader.

And if you are reading this line after the other lines, you have at least read one page of my thesis. Thank You.

Pascal Mellema

February 2020, Noordwijkerhout

Abstract

This thesis proposes patterns of use for semantic browser interface specifications for inferred data using the Fresnel semantic browser style language. We also propose our own extension called Fresnel Inferences. The use of style for the presentation of inferencing, independent of other styles for presentation, contributes to the flexibility, reusability and development of information systems. We map references to the user interface with style sheets, which can then be adapted by an administrator to specific business needs through inferencing of display classes. This research proposes both use of current and extensions to form-based interface specifications for Fresnel to display presentation of inferencing results to end users in a human understandable way. Displaying the inferencing using styling and color in semantic browser as user interface components, the usability of Fresnel is further extended.

For this thesis Fresnel code is used for repeatability. Mockups show how a conforming browser with SMW as platform could implement it. In addition, it provides both the ability to make information of existing data available and form-based annotation interfaces to new data. The widely used rule-based fictional case of a car rental company EU-Rent is used as a demonstration how the mapping can be applied.

Key terms

Fresnel, Protégé, CSS, EU-Rent, Semantic Web

Contents

1. Introduction

1.1.Introduction

Semantic Web (SW) is a vision of W3C of the Web of linked data. The Semantic Web technologies like Resource Description Framework (RDF), RDF Schema (RDFS), Fresnel and Web Ontology Language (OWL) make it possible to create data stores on the Web, build vocabularies, and write rules for handling data. Fresnel provides an RDF vocabulary intended to encode information about how to present SW content to users (W3C, 2005). The existing style language Cascading Style Sheets (CSS) (W3C, 2019) is for describing the presentation of documents, including colors, layout, and fonts and not data. CSS is used by Fresnel, CSS hooks for an external CSS-like language to manage the aesthetics, to determine the actual appearance of the data (Pietriga, Bizer, & Karger, 2006).

This research aims to present a model to show how current Fresnel can be used, how to extend Fresnel with inferred data presentation that can be used as user interaction component by semantic browsers in compliance with the Fresnel format RDF information. How this model works and which insights it provides is discussed. We explore to what extend the current options of Fresnel to present inferred data is feasible.

OWL Wiki Forms (OWF) utilizes the SW ontology editor Protégé and the MediaWiki extensions that map SW ontologies to Semantic Forms-based semantic wikis (Rutledge, 2013). An OWF project is Fresnel Forms (FForms), a Protégé plugin for producing form-based semantic wikis for given ontologies (Rutledge et al., 2016). Style specification can be modified with the Fresnel editing options. Thus, OWF and FForms makes semantic wiki interface creation more efficient by generating a default interface for given ontologies (Rutledge, 2013).

The EU-Rent case study has been implemented in SW (GEIST Research Group, 2015b) technology and is used as an example for the rules used to derive a business rule. For this research an example case will be used in part based on the widely used EU-Rent.

1.2.Motivation/relevance

Studies that are earlier done at our faculty have examined business rules in the Semantic Web (J. Bos, 2018), (Slootweg, 2016). Another study implemented Page Forms with Semantic MediaWiki (SMW) by generating Fresnel-defined styles from ontologies to build information boxes (Rutledge et al., 2016). This research pays attention to clarity for domain experts, the humans, and recognizes that the ontology has to be understand by machines. This includes reasoning about RDF-defined triples, with inference logic ("rules") defined by SW technologies. Which inferences are valid, given certain patterns of triples, is defined by RDFS and OWL standards. RDFS handles linking classes to subclasses and properties to classes. OWL also builds up with rules for describing classes on the basis of permitted values for properties. All these standards differ in the derivation that they support, but all use the idea of inferencing to describe the meaning of a model (Allemang, 2008).

This research aims at contributing to Fresnel by investigating how and to what extend inferred rules based on Semantic Web can be used in to provide information to the end user.

The main question of this research is:

"How can patterns of use of interface style for data that is inferred on the Semantic Web using Fresnel be specified or extended?"

Specifying their presentation by using and extending Fresnel and related styling definitions shows the user if data is inferred based on the RDF-defined triples. This is done in this research by identifying inferred business rules in the Semantic Web and visualizing the inference in Fresnel semantic browsers.

The research question is broken down into the following sub questions:

a. How can current use of Fresnel contribute to interface style for inferred data in a Semantic Web browser?
b. How can possible extension to Fresnel to interface style for inferred data in a Semantic Web browser be defined?
c. How can patterns of current use with extensions work?

To validate and evaluate the results for our research questions, examples are taken from Protégé (Stanford Center for Biomedical Informatics Research, 2016). Protégé is one of the most used Semantic Web tools supporting OWL 2 and RDF specifications from the W3C. It provides styled displaying information for interactive navigation of ontology relationships. Protégé has a plug-in architecture that can be used to build ontology-based applications. The output of Protégé can be integrated with rule systems. We use Protégé as a model of an existing interface that shows inferencing to the end user using style. The displaying information as presented to the end user in the user interface environment in Protégé is used as a basis for our model to create an analogy to validate and evaluate the Fresnel user interface displaying definitions, as defined as a result for research question 'a.' and 'b.' mapped to our example ontology and defined as a result for sub question 'c.'.

Our evaluation questions are derived from the questions stated above:

i. To what extent can interface style in a user interface definition for data that is inferred on the Semantic Web from an example ontology editor, be defined?
ii. What extensions to Fresnel can be defined to contribute to interface style for inferred data in a Semantic Web browser?

Our research questions and evaluation sub questions will be addressed in our model, of which the design is detailed in Section 3. The described literature study as described in Section 2 explores the research domain and examines existing models, tools and examples to be used for our model. The results are described in Section 4. The final conclusions are described in Section 5.

1.3. Terms of reference

The Protégé ontology editor, the plugin FForms is used that helps create Fresnel-defined interfaces and then exports them as semantic wikis (Rutledge et al., 2016). With FForms, ontologies are processed into triples in accordance with the Fresnel vocabulary for semantic browser displays (Rutledge et al., 2016). This means that FForms processes the currently active ontology in Protégé and exports this as a semantic wiki for that ontology. And as last step, FForms exports Fresnel-defined interface as a SMW. SMW provides both the ability to browse existing data and form-based annotation interfaces to add new data. As an example of business rules, a part of EU-Rent was used as a model to demonstrate the mapping.

1.4. Main lines of approach

We propose code for Fresnel that applies possible lens, sublens and related styling parameters to present displaying information of data that is inferred on the Semantic Web to the end user in a human interpretable way. Such as a change of background color in a user interface environment, but also for displaying the explanation information of the related inferencing, such as color, font and border. For code suggestions and expected output displaying information presentation, we use mockups to present possible implementation outcome for the user interface environment.

2. Theoretical framework

2.1. Research approach

The purpose of this research is to provide answers and to propose a model for displaying information in a user interface definition regarding data that is inferred on the Semantic Web in a SW browser environment by using Fresnel. Our theoretical framework is based on the following different types of consulted sources: peer-reviewed scientific articles from magazines and conference proceedings that contain scientific information, scientifically supported websites for more general information about software, implementations and online standards and other general sources.

For this study, a literature review is performed to answer the central question as stated in 1.2. The primary goal of this literature review is to develop a sound base of knowledge and understanding of relevant research and the trends that emerge from that research (Saunders, Lewis, & Thornhill, 2007). Research questions are formulated to help answer the central question and to provide new insights (Saunders et al., 2007).

The subsequent research questions, based on the central research question, are formulated as follows:

- What inferred data can be created with technology of Semantic Web?
- How can results from inferred data with technology of Semantic Web be explained?
- What are Fresnel rule styles?
- How can Fresnel be extended for usage of rule styles?
- How did other recent research evaluate rule styles?

With answering these research questions, the theoretical framework will be created. Based on the theoretical framework a hypothesis will be formulated, to be used in the remainder of this research.

The next paragraph describes implementation of the literature review. The last paragraph describes follow-up of the research.

2.2. Implementation

The displaying of inferred data using style is supported by software tools like Protégé (Stanford Center for Biomedical Informatics Research, 2016). User interface components show the end user inferencing with styling options to describe its functional specifications for business rules.

An OWL 2 implementation of EU-Rent has been created as part of the research to map Semantics of Business Vocabulary and Business Rules (SBVR) to OWL 2 (Reynares, Caliusco, & Galli, 2014). In this research there are several applications for such a mapping presented. The use of ontology reasoners to prove the consistency of business domain information, the generation of an ontology intended to be used in the analysis stage of a software development process and the possibility of encapsulate the declarative specification of business knowledge into information software systems by means of an implemented ontology are considered most important. The SBVR version of EU-Rent that was included as a supplement to the SBVR 1.0 specification (Object Management Group, 2008) as the source. Stating concepts, relations and necessary rules for new lens and format specifications of Fresnel serves as a basis for our model to demonstrate our aimed displaying representations of inferred data on SW in a user interface environment (Reynares et al., 2014).

2.2.1. Inferencing with Semantic Web technology

The displaying of RDF data by SW browsers with the primary intend to present information for machine consumption in a human-readable way (Pietriga et al., 2006) can be used in our aim to present displaying information of data inferred on the SW to end users. In SW, things can be inferred based on facts that are stored in RDF triples. All RDF statements or RDFS and OWL are First Order Logic and Descriptive Logic to facilitate inferencing. Where inferencing is a systematic process of adding new tuples to an RDF graph based on rules.

2.2.2. Fresnel rule styles

Fresnel was introduced as ontology for the presentation of data from SW ontologies. Where CSS determine how XML documents should appear on web browsers, Fresnel can be used to add style information to information platforms based on semantic ontologies (Pietriga et al., 2006).

Fresnel's presentation knowledge is declaratively expressed in RDF and is based on two concepts: 'lenses' and 'formats' as depicted in Figure 1. A lens specifies which characteristics of the RDF data are shown and how they are ordered. A format defines the format of the content and any links with external CSS that can be used to style the output (Pietriga et al., 2006).

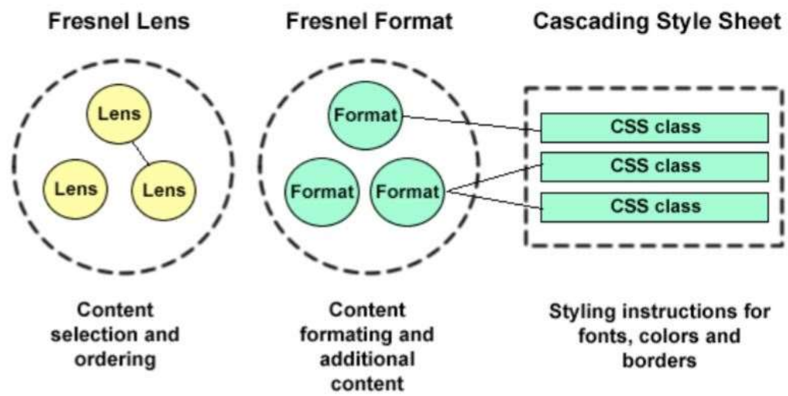

Figure 1: Fresnel foundational concepts (Pietriga et al., 2006)

Lenses indicate which properties of the RDF resources are shown and how these properties are arranged. Formats indicate how the content selected by lenses is formatted and can also generate additional static content and hooks in the form of CSS class names that can be used to output via external CSS (Pietriga et al., 2006).

Looking at other studies at the Open University in the Netherlands that are being carried out can help answer the question about how Fresnel can be used to format the default style for a given ontology when there is inferencing. The research done by Rutledge (Rutledge et al., 2016) is a mapping from any ontology to Fresnel style data and from Fresnel data to form-based semantic wikis, but not for inferences. Only data structure is processed, and only to make simple form input structure (Rutledge et al., 2016). Bouwer proposes business rule styles, alethic and deontic, in interfaces by extending Fresnel with Fresnel Violations (Bouwer, 2019).

2.2.3. A prototype for rule styles based on Protégé

The use in our research of an EU-Rent case ontology example for displaying inferencing information in the Protégé user interface environment provides the needed displaying information to support the creation of an analogue in SW user interface environment for inferred data presentation to end users. Presenting the difference in the Protégé user interface environment between data and inferred data should be feasible to automate, based on URL namespaces in the triples of the explanation.

The displaying inferencing information in the Protégé user interface environment notifies the end user that there can be information derived from this part of the interface providing in-depth information on the inferred triples. This includes the triples for the particular rule, along with the data instance triples that trigger the rule.

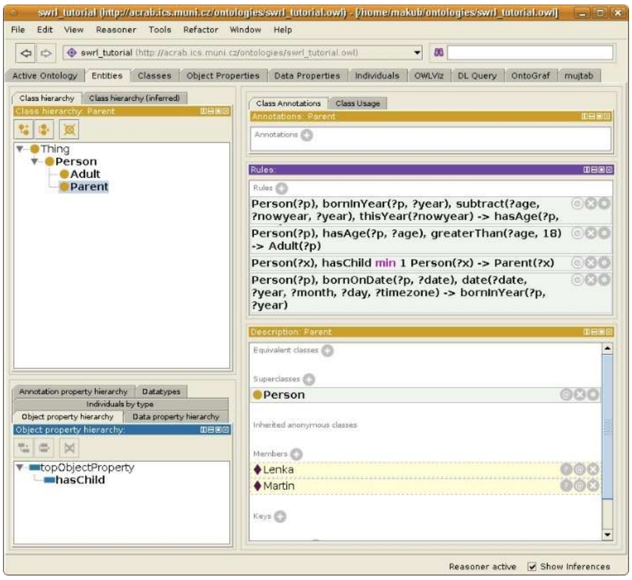

Figure 2: Protégé-inferred triples (Kuba, 2012)

Protégé shows inferred triples with dotted border and background color as shown in Figure 2. An analogue of this presentation of an additional information window is aimed at as part of our model to display information styles in the form of an explanation box containing explanatory information on inferred triples in SW user interface environment.

2.2.4. Presentation information of inferred data using Cascading Style Sheets

Cascading Style Sheets are used for defining how HTML documents should be presented to end users on web browsers. HTML uses predefined tags which means that for each tag the meaning of, and how to display it is well known by system administrators. Cascading means that a specific stylesheet can be built on a more general stylesheet. Web browsers use a default general CSS when no other stylesheet is present for presentation of HTML documents. Within the field of model-driven development CSS is used as an example to develop in a similar way interfaces for data systems (Rutledge et al., 2016).

2.2.5. Presentation information of inferred data using FForms, Page Forms and SMW

FForms is (Rutledge et al., 2016) as a plugin in Protégé allowing the user to generate an interface that can be stored as an RDF file of Fresnel triples. FForms is created with the functionality that allows for exporting the system and interface excluding data. The result is a wiki with the possibility to enter data and to post the data on the Semantic Web. We use the plugin FForms in Protégé to generate an interface that can be stored as an RDF file of Fresnel triples. The output from FForms can be wiki text for SMW but can also be Fresnel code. FForms uses default styling and allows for customized styling (Rutledge et al., 2016).

By import of the XML file, in a wiki, the Page Forms assist the user to enter data (Rutledge et al. 2016). We use FForms for our model as a means to create Fresnel code as a basis as described in Section 4. We use Page Forms and SMW as a representation for our intended displaying information environment for data inferred on the Semantic Web. Our work examines the use of FForms to speed up our implementation process.

2.3. Results and conclusions

Little literature is found on creation and specification of displaying information specifying interface style for data that is inferred on the Semantic Web. Most related work is aimed at business rules and presentation of related information, rather than style information for displaying information. Protégé shows inferred data with a different color background and dotted border style as displaying information in the user interface environment. Secondly, Protégé provides additional information presented in a separate window displaying an "explanation" for a triple. The displaying of information in the user interface environment of Protégé provides us with information we use for our model to present displaying information to end users for data inferred on the Semantic Web.

Fresnel expresses RDF presentation knowledge in a declarative way and Fresnel applications should take all formatting information into account but are free to interpret and adapt them in a way that is appropriate. Fresnel does not dictate how to transform the tree resulting from the formatting step into an actual target document of the chosen output format (W3C, 2005) using styling hooks to CSS. Styling can be added by using (X)HTML with CSS to generate an output document like (X)HTML containing inline style attributes.

The EU-Rent case has partially been implemented in Semantic Web (Reynares et al., 2014) and also in Ampersand (H. Joosten, 2014). Part of the EU-Rent case containing concepts, relations and business rules will serve us as an example ontology in Protégé we can extend. We use this extended ontology for our model during this research. The EU-Rent case does classify rules using an enforcement level. This means that these rules can be verified or inferred. This inference can then be used in a Fresnel lens to trigger a (form-based) interface.

Research is conducted at the Open University in the Netherlands regarding the usage of business rules and Fresnel. Bouwer (2019) proposes business rule styles, alethic and deontic, in interfaces by extending Fresnel with Fresnel Violations. The use of a rules-based approach to adapt the user interface to changes in business rules can affect management information systems (Bouwer, 2019). Taking the existing work found into consideration for this research, it can be concluded that it is feasible to contribute by proposing a style specification that shows the inference triggered in Fresnel and discover what influence that has on style.

2.4. Objective of the follow-up research

For this research the next step is to describe our model as a partial case based on the EU-Rent case. We use an implementation of triples involved in making the inference triggered in Protégé to serve as a representation of displaying information for our model. We use this obtained information in Fresnel to answer our questions stated in Section 1. Our research contribution to Fresnel lies with the fact that inferred triples need to be obtained with displaying information. Our research model is aimed to answer our research questions regarding displaying information of data inferred on Semantic Web to end users in a Semantic Web user interface environment. A Semantic Web user interface environment used in other research, such as the work done by Rutledge (Rutledge et al., 2016), Slootweg (Slootweg, 2016), Van Gysel (Van Gysel, 2017), Bos (J. Bos, 2018) and Bouwer (Bouwer, 2019) is SMW. We follow their intentions of using SMW browser environment in our research for the purpose of displaying information to end users.

3. Methodology

In this section we explain our research approach and elaborate on the method we use to develop a model as well as patterns of use of Fresnel for displaying information for inferencing. We create our model by analyzing and comparing the presentation of displaying information for inferencing in the Protégé user interface environment by including extensions to Fresnel and testing examples of these. Our goal is to create an analogue of found displaying information for data that is inferred on the Semantic Web using Semantic Web technologies. The Protégé plugin FForms is used to obtain Fresnel code and SMW browser serves as our intended displaying information environment. We use an existing ontology derived from EU-Rent case to create the inferencing for displaying information of inferred data in Fresnel. From this, alternative displaying information is derived and added to Fresnel as our scientific contribution. How well our model facilitates the current state and how it facilitates changing is discussed in Section 5.

In 3.1. the research design is presented and then the technical design in 3.2, followed by the data analysis in 3.3. This Section is concluded with the reflection on validity, reliability and ethical aspects of the research.

3.1.Conceptual design: select the research method(s)

To answer the question of this research, the form of deduction is chosen, and a model is designed based on the literature study. This model is verified by applying it to the EU-Rent case. The mapping of business rules to a Semantic Web interface that can be implemented in Fresnel and its extensions. Protégé, Fresnel (Forms), OWL, RDF(S) and SMW are used components as Semantic Web 'inferencing' technologies. The Semantic Web technologies, as RDF(S) and OWL are used to describe concepts and relationships within data for humans and machines. SW models expressed in RDF(S) and OWL drive user interfaces and provide end users a human understandable presentation of data by displaying information using styles in the user interface environment.

Fresnel is developed as an extendible ontology to add style information to information platforms based on semantic ontologies. Fresnel can also be extended for more specialized needs, like our extension for style on inferred data (Pietriga et al., 2006). In our research Fresnel serves as a bridge between Protégé and SMW.

Based on our model the following hypothesis was formulated:

How and by what means can a style specification show results from inferencing and links to the Semantic Web ontological components involved in making the inference triggered in Fresnel.

This formulated hypothesis is the empirical part of the research and will be used during the rest of this work.

The EU-Rent case is made to be a common reference point for comparison in research. We use the EU-Rent case as a starting point in Protégé for our model. Although EU-Rent is a fictional company developed for research purposes it provides a close enough resemblance of a real existing company. Research projects are based on EU-Rent and the case is well documented (Object Management Group, 2008). Other research currently done at the Open University of the Netherlands investigates possible implementation of alethic and deontic rule in the user interface (Bouwer, 2019) using the EU-Rent case. We perform with our research a single iteration within the realm of design science based on the model of Alan Hevner (Hevner, 2007).

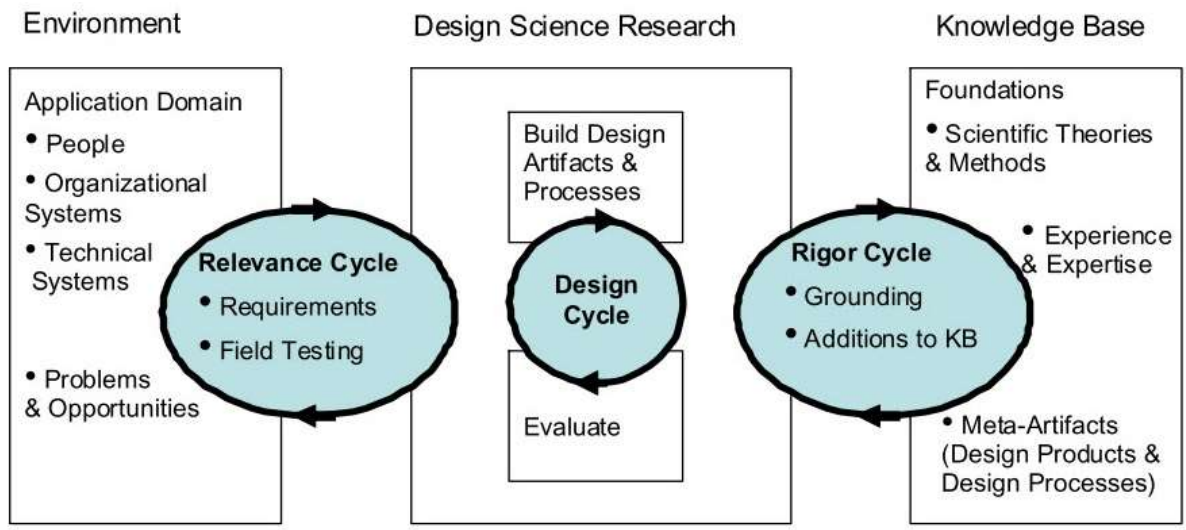

Figure 3: Design science research cycles (Hevner, 2007)

The design science model of Hevner (Hevner, 2007) shown in Figure 3 consists of three cycles. The Relevance Cycle to bridge the contextual environment of our research with the design science activities. This is executed in Section 1 where we describe the relevance of our research. The Rigor Cycle to connect our activities with the knowledge base of scientific foundations, experience, and expertise where our research is based upon by performing a literature research as presented in Section 2. We address the central Design Cycle in Section 4 using a descriptive evaluation method. In the Design Cycle we iterate between the core activities of proposing additional code to support our model and provide insights in how our model works with the use of current SW technologies and possible extensions to test our theory how and by what means displaying information for data that is inferred on the Semantic Web using SW technologies style can be implemented. We explore different styling possibilities for displaying information for inferred data using SW technologies to extend our knowledge and gain understanding in the presentation of displaying information, while we take into account possible unexpected challenges as the research progresses.

Where necessary, the data can be refined or otherwise improved to suit the needs of our model. During the execution of the research, a displaying information style implementation of the inferred data will be created to serve as a representative ontology from where the inferred data is derived. The created styles will partly be imitated from the visual representation on inference that is shown in Protégé. Protégé serves as a target for our model for displaying information of inferred data in a Semantic Web user interface environment. We conclude in Section 5 with our evaluation of our proposed model and processes of the research.

3.2. Technical design: elaboration of the method

We present a schematic overview of the steps that will be carried out during our research in Figure 4. We take into account that as the research progresses, previous steps may need to be re-examined in order to be adjusted. Our presented steps are part of a single iterative process to bring the desired result closer to discovery to provide insight with useful information about understanding, conducting and evaluating results of our research. The separate steps can be interpreted as mini-iterations of our main iteration process. The steps are evaluated during the process.

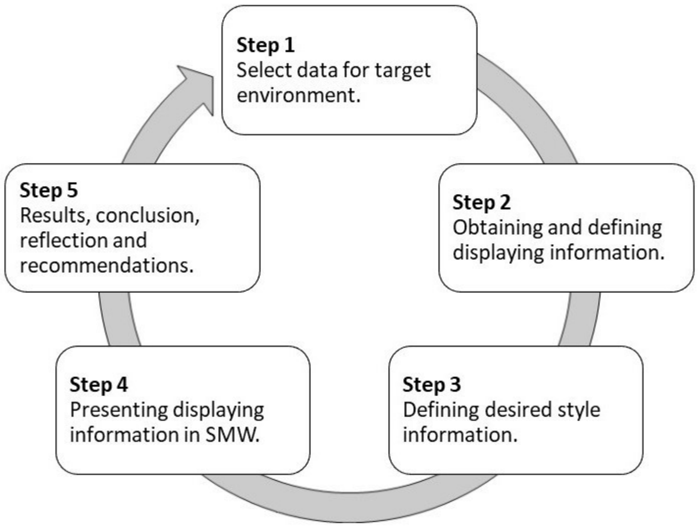

Figure 4: Schematic overview research steps

Step 1: Select data for target environment

Data must be created to demonstrate the displaying information of inferred data in our target environment Protégé. Based on the literature review done for this research the EU-Rent example ontology will be used as part of the source data for our model. This is the first step.

Step 2: Obtaining and defining displaying information

In the second step, data is selected and can be enriched with data to obtain the inferencing displaying information from the user interface of Protégé. Protégé notifies the end user that there can be information derived from the part of the interface providing in depth information on the inferred triples. This includes the triples for the selected rule, along with the data instance triples that trigger the rule. Rule versus triggering data has an influence on the interface, and how it gets styled. The influence on the interface, and how it gets styled is of importance because this will form the basis of our model.

Step 3: Defining desired style information

During the third step, the main contribution will be made to answer the main research question to extend the use of Fresnel by creating a Fresnel (Sub)lens (selector) that responds to an inferencing rule derived from SW technologies by replicating the influence on the interface in Fresnel. This includes the triples for the particular rule, along with the data instance triples that trigger the rule. Seeing the difference should be feasible to automate, based on URL namespaces in the triples of the explanation. The selection has to be made based on the displaying information style presented for an inferred rule as is presented to the end user in the user interface of Protégé.

Step 4: Presenting displaying information in SMW

The fourth step is to present displaying information for the end user in SMW. Style specification can be implemented for showing which triples are inferred and how to show/link to their triggering data and how to show/link to the rule(s) triggered.

Step 5: Results, conclusion, reflection and recommendations

In the fifth step, our results of the created inferred style displaying information described in the previous steps is discussed. The results of our implementation are compared to the displaying information as presented in the user interface environment of Protégé. We also present the conclusions, reflection on the results, the process of the executed research and provide recommendations for future research.

3.3.Data analysis

A target interface model is created for the mapping by using Protégé. By defining the mapping from Protégé to Fresnel the modelling is done. At the mapping stage the EU-Rent example ontology and Fresnel, FForms and any extensions that may be required during the creation of the interface model and mapping may be adjusted if further extending existing code is required. A comparison is made between the results in user interface environment displaying the inferred data style presentation between SMW and Protégé to validate the mapping and interface model.

3.4.Reflection regarding validity, reliability and ethical aspects

This part describes an explanation of the chosen research method and identifies potential risks. The validity is determined on the basis of internal and external validity (Saunders et al., 2007). The internal validity does not apply to our exploratory research. For this research the external validity is of importance because it contributes to the development of Fresnel. However, it should be noted that other types of business rules could lead to new requirements for a rule style language.

In this research are publicly available implementations used. Changes to existing implementations and the prototype have been fully documented and have been demonstrated exclusively by using open source tools. In this way, it can be ensured that the research is repeatable and consistent (Saunders et al., 2007).

The data used for the execution of this study are mainly derived from the EU-Rent case. EU-Rent is fictitious and will therefore not pose a risk for data protection. All resources used for developing the prototype are open source. This ensures that the research is reproducible. Related work on which the research is based is stated at the place of application and in the references. General sources on which the knowledge is based are also included in the reference list.

4. Results

In this Section, the case results are discussed. Our model addresses the research question formulated in Section 1. An overview of the applications used for our ontology addition named 'finf', created for our model to style inferred data is provided in paragraph 4.1. Default style for inferred triples, for which we introduce a new component for Fresnel that recognizes if a triple is inferred. is discussed in paragraph 4.2. We present and discuss an option to alter the presented displaying style for specific inferred data in 4.3. Prototyping info box we name DetailsBox adds meaning and message to be able concerning the inferred triple displayed, is discussed in paragraph 4.4. Each paragraph within this Section contains the conclusions for the sub question discussed.

4.1. Overview

For this research, the EU-Rent case is used to support our model of visual representation of the influence of the inferred data in Semantic Web browser interface. Data must be created to demonstrate the mapping of EU-Rent. The demo is an evaluation of our proposed technique. Our goal is proposing and arguing for this technique. The EU-Rent case we use for setting up the prototype is the ontology created by Reynares (Reynares et al., 2014) as an SW encoding of Annex G of the EU-Rent case (Object Management Group, 2008). The applications and their respective versions used for this research can be found in Table 1. Code fragments in boxed examples are in `Courier New` font and our own added code is in **`Courier New Bold`** font. Our ontology code parts that can be found in this section are collected in Appendix A of this thesis.

Table 1: Applications used

Application	Version	Additional information
Protégé	5.0.0	Build beta-17
Fresnel Forms	2.0.0	Plugin for Protégé
HermiT Reasoner	1.3.8.3	Plugin for Protégé
EditPlus	4.1	Text editor and syntax highlighting

We use an example ontology named CarMovement.owl based on the EU-Rent case (Reynares et al., 2014). We use CarMovement.owl ontology and include this ontology in our ontology addition named 'finf' created for our model to how and to what extend the appearance of inferred data can be manipulated. We create our addition 'finf' to the existing CarMovement.owl ontology for making inferencing explicitly visible in Semantic Web browser interface.

4.2. Inferencing styling options

FForms generates Fresnel definitions and uses SMW as implementation platform (Rutledge et al., 2016). We use FForms in our model to provide a standard output to obtain the information Fresnel can provide to the end user for our inferred data displaying style by use of Fresnel and CSS.

4.2.1. Inferencing style in Protégé

In this step in our research, we create inferred data for our contribution to Fresnel using the CarMovement-finf.owl ontology to obtain the displaying information of the inferencing using style in a semantic browser as user interface components. In Protégé, the class hierarchy is displayed as a tree. We add two instances to our ontology named 'Car_A' and 'Movement_X'. We use underscores for spaces following the usage of underscores for spaces as used in CarMovement.owl. After we added 'Car_A' to our ontology, Protégé shows the asserted triple 'Car_A' 'has_destination_Branch' 'LAX_Airport_Agency' as a property assertion for 'Car_A' without any reasoner being activated. The explicit relationship that was stored is ‘Movement_X has_rental_car Car_A’. In the panel Property assertions, the inferred data is displayed in Protégé when reasoning occurs as shown in Figure 5. In our case, the inferencing Protégé presents in the user interface is an inverse relationship between the individuals ‘Movement_X’ and ‘Car_A’, ‘Car_A is_assigned_to Movement_X’.

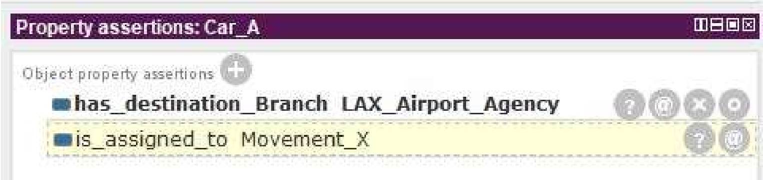

Figure 5: Property assertions: Car_A with inferred triple in Protégé

The style specification we implement demonstrates the use of style specification for inferred triples has an influence on the interface of Protégé by presenting a yellow background color and a dotted line, see Figure 5. Our focus is on the background color and border style for inferred data. Our proposed mapping of the style defined in CSS for the presentation in SMW is shown in Table 2.

Table 2: Derived styling from Protégé in CSS

Protégé UI presentation	CSS (property: value)
Yellow background color	background-color: yellow
Dotted outline	border-style: dotted

We examine how the presentation in the interface of Protégé of inferred data is included in the application by entering the menu. We could not find any options or settings in the menus of Protégé to alter the interface presentation of the notification of inferred data formatting in the user interface of Protégé.

4.2.2. Inferred triple recognition in Fresnel

We introduce a new component for Fresnel that recognizes if a triple is inferred or not. If a triple is inferred, style can be applied to display the distinction to the end user. For our model, we need to know if the triple is inferred. Protégé provides information on which triples infer a particular triple. It is empty if the triple is not inferred. Protégé makes a distinction between asserted and inferred triples. Every part of an RDF triple is individually addressable via unique URIs. Protégé uses this information in its user interface by showing individual inferred triples with distinct style. This information can thus also be made available for Fresnel with the same technology Protégé uses. Our finf ontology with the namespace prefix ‘**finf:**’ as shown in Example 1:

```
finf:Inferred        rdf:type          rdfs:Class .
finf:InferredLens    rdfs:subClassOf   fresnel:Lens .
finf:infSelector     rdfs:subClassOf   fresnel:valueSelector;
```

Example 1: finf extension ontology OWL code

We propose to extend Fresnel propertyFormatDomain with a valueFormatDomain in a similar step of abstraction that propertyFormatDomain takes from class selectors. In CSS, selectors are patterns used to select elements you want to style. The .class selector selects elements with a specific class attribute (W3C, 2019). Style specifications are declared by properties and related values. We show our proposed inference selection in Fresnel as shown in Example 2.

```
rdfs:domain          finf:Inferred;
fresnel:instanceLensDomain “SELECT ?value
WHERE {?value RentalCar:is_assigned_to ?CarMovement .
?CarMovement rdf:type finf:infSelector }” ^^fresnel:sparqlSelector .
```

Example 2: Specification of proposed inferencing selection in Fresnel

In fresnel we select the inferred triple using our SPARQL query, like selecting an element to style. We select a specific `finf:valueFormatType` ‘`inferred`’ and append a selector, our `fresnel:styleClass` ‘`finfInferred`’. The property links the CSS class ‘`.finfInferred`’ for reference from a central CSS.

We connect our declared class to our referenced CSS, where we use the .class selector as our reserved class name ‘`.finfinferred`’. This template-based approach provides the end user with a proven method to apply style to inferred triples. We introduce a `finf:valueFormatType` property for `fresnel:Format` resources that has as value either "inferred" or "asserted". The value of the format `Fresnel:valueStyle` provides the style for inferred values.

For our Semantic Web UI displaying representation of inferred data, we use existing Fresnel extended definition to refer to our reserved CSS class named ‘`.finfInferred`’ used to apply appropriate styling definitions. We use an example stylesheet we named ‘finf.css’ for evaluation and illustration for our work where we define our displaying properties so that it contains the necessary classes, as shown in Example 3.

```
:finfGroup  rdf:type fresnel:Group ;
 fresnel:stylesheetLink
<http://is.cs.ou.nl/OWF/index.php5/Masters_Thesis_Pascal_Mellema/finf.css> ;
```

Example 3: Reference to external CSS in Fresnel

We extend the behavior of the Fresnel end system semantic browser by having it insert our reserved class in the HTML element as attribute ‘`class=finfInferred`’. We show that our extension is valid and create the required code in Fresnel based on the work done by Rutledge (Rutledge et al., 2016).

```
:finfInferredLens
        rdf:type                       fresnel:Lens ;
        fresnel:group                  :finfGroup ;
        fresnel:hideProperties         () ;
        fresnel:purpose                fresnel:defaultLens ;

:finfInferredFormat                    fresnel:Format ;
        find:valueFormatType           "inferred";
        fresnel:valueStyle             "finfInferred"^^fresnel:styleClass  .
```

Example 4: Proposed specifications in Fresnel

The abstract box element from Fresnel (Pietriga et al., 2006) we use for our ‘finfInferred’ in FForms interface is an instantiation of the Fresnel abstract box model named Value Box. This Value Box contains one property value, which we labeled with our reserved value ‘finfInferred’. Our inferred data property value is displayed using a sublens. Each lens gets a group in Fresnel, in that group the style definition of that lens is defined.

We edit Fresnel code using FForms in Protégé. After the save/export from FForms, we examine the information related to the triple ‘Movement_X has_Rental_Car Car_A’. However, we did not discover any information related to the inferred triple ‘Car_A is_assigned_to Movement_X’. This would provide us with hooks to apply specific style to inferenced triples. As we found, the Fresnel style cannot know that certain properties in one direction will always be inferred. Fresnel cannot determine if triples are inferred, only the end system knows.

Our examination of the FForms interface shows options to include styling options for four defined Fresnel abstract boxes (W3C, 2005) from the Fresnel abstract box model. FForms interface allows for styles to lenses and triples. Class names are used in Fresnel as selector. This can be compared to element names that are used as CSS selectors. A lens selects a class that displays all the RDF-resources in that class. We did not discover options to determine if triples can be distinguished from each other.

We propose to extend FForms interface to display and include inferred triples making it easier for users to edit style for inferred triples after reasoning is done. Secondly, we propose an extension to the FForms interface to include a reserved CSS class name ‘**`.finfInferred`**’ which tells CSS that this value, and thus triple, is inferred. This is a fixed, standard class that all implementations of our extension will recognize. Reference to a default external CSS stylesheet provides the end user external links using CSS styling instructions associated with CSS classes in a human readable way. For the use of FForms in Protégé, we suggest a Fresnel inferred triple style for inferred triples in the user interface as shown as a mockup in Figure 6.

Figure 6: Mockup FForms inferred sublens extended with inferred triple

We propose that any tool conforming to our extension, exports information of that inferred ontologically structured properties, then these can be accounted for in the generated user interface. In Fresnel, the default ‘showProperties’ gets the inversed property. We assume that this can be done if a tool like FForms can fill sortProperties for the default Fresnel by querying for the inverse of the property. Should both inverse and regular properties be included then the template and form on both sides could get its own place for its end of this "two-way" inversed property. One convenience of an inverse property is that it would be possible to provide each direction with its own `rdfs:label`. In addition, in this user scenario, the designer could add inverse properties to be able to add such labels for incoming links as a style-like activity.

We examine where the style in the generated Fresnel output is included by adding ‘finfInferred’ in the interface of FForms to the right mouse menu ‘edit CSS’, see Figure 7, for the ‘has_Rental_Car’ triple.

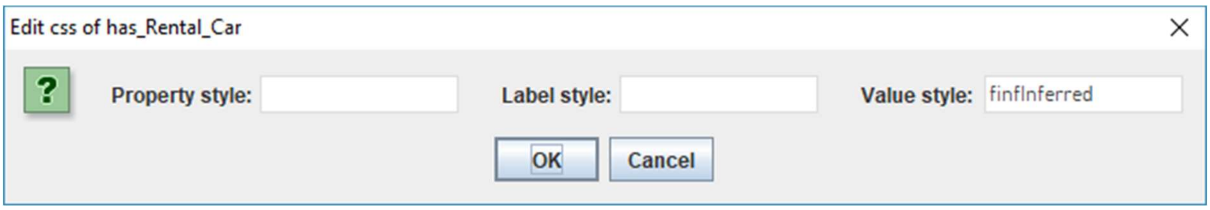

Figure 7: ‘ValueStyle’ in FForms

In the Fresnel code generated by FForms we find that ‘ValueStyle’ is added to the ‘CarMovement:has_Rental_Car’ format as a Fresnel style class, defined as: ‘`fresnel:valueStyle` **`"finfInferred"`**`^^fresnel:styleClass`’. In the same way we create a style class in Fresnel for our inferred triple as presented in Example 4. In our example the object property ‘`is_assigned_to`’ is the inverse of ‘`has_Rental_Car`’. Since only the end system knows what is inferred, Fresnel style cannot know that certain properties in one direction will always be inferred.

4.2.3. Inferencing style in SMW

We propose to add our reserved CSS class '`.finfInferred`' to a central CSS the browser uses in the active Semantic Web browser interface to make inferred data distinctive for end users. The `fresnel:valueStyle` property links the CSS class '`.finfInferred`' for reference from a central CSS We examine where the style in the generated wiki output is included by adding 'ValueStyle: finfInferred' in the interface of FForms to the right mouse menu 'edit CSS', as previously shown in Figure 7, for the 'has_Rental_Car' triple. We examine the generated output and expect to find an element with an attribute for our reserved CSS class '`.finfInferred`'. We found a div element containing a predefined class attribute named '`ib_value`' and a style attribute containing '`finfInferred`' as shown Example 4.

```
<div class='ib_value' style='finfInferred'>
   (W3C, arraymap:{{{owl-Thing-CarMovement-has_Rental_Car|)}}|,|xxx|[[owl-Thing-CarMovement-has_Rental_Car::xxx]]}}
</div>
```

Example 4: Class attribute (in red text) and style attribute (in blue text) from FForms via XML MediaWiki export

We re-examine the Wiki XML to discover where our style 'finfInferred' we included in the FForms interface is defined. We found that our included style 'finfInferred' did not convert to a CSS class as expected but is included as a CSS style attribute to the div element. After further examination, we found four default CSS classes defined in Fresnel which FForms includes on export. These are: ib_resource for resource style, ib_property for property style, ib_label for label style and ib_value for value style.

We propose to extend FForms ontology to include our reserved class 'finfInferred'. The combination of our previously proposed definition for inferred triples and the alternative class definition inclusion makes it possible to export and style inferred data in CSS for use in SMW.

For our `fresnel:valueStyle` '`finfInferred`', we add a CSS class named '`.finfInferred`' to our connected finf.css to apply our defined style as shown in Table 2. The output Wiki XML for our SMW presentation is presented in Example 5 as example code. We include our example code and a reference to finf.css for presentation in SMW browser.

```
<div class='finfInferred'>
   {{#arraymap:{{{finf-Car_A-is_assigned_to |}}}|,|xxx|[[finf-Car_A-is_assigned_to::xxx]]}}
</div>
```

Example 5: Class attribute named finfInferred (in red color)

Van Gysel (Van Gysel, 2017) has provided possible solutions to use generated data in SMW to present displaying information to the end user. We show our inferred triple in the Semantic Media Wiki browser interface with our proposed styling using our reserved class value '`.finfInferred`' in our extension of Fresnel for `fresnel:valueStyle` as shown as our relevant CSS code part.

By presenting a default class for inferred data in Semantic Web browser interface displaying style and using this style for inferred data presentation as shown in the mockup in Figure 9, the end user is provided with human understandable information to distinguish the inferred data from other information presented in the SMW default browser interface.

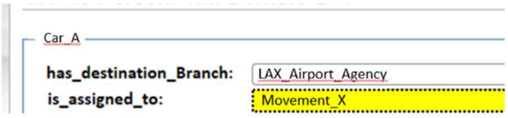

Figure 9: Mockup inferred triple Car_A-is_assigned_to with class Inferred in SMW browser interface

We add our reserved CSS class '**.finfInferred**' to the CSS the browser used in the active Semantic Web browser interface to make inferred data distinctive for end users. In SMW, the CSS named MediaWiki:Common.css is one of the global stylesheets and is appended to the hardcoded stylesheet of MediaWiki. The MediaWiki:Common.css is appended throughout the site making it a logical place in SMW for a default inferred. In general, a piece of CSS code can appear in many places. Our addition of CSS code is:

```
.finfInferred {
            border-style: dotted !important;
            background-color: yellow !important;
        }
```

Figure 10: Our reserved class .finfInferred in CSS code

We use the code `'!Important'` to override other styles. With our proposal we also extend the possibilities for further developments on the Semantic Web.

4.2.4. Conclusion

We proposed a new component for Fresnel that recognizes if a triples are inferred by introducing **finf:valueFormatType 'inferred'**. Making it possible for end users to apply style to inferred triples. As an extension to FForms, we propose detailed information to the end user and providing possible options to export inferencing information to Fresnel. As a new reserved name, we present '**finfInferred**'. We presented a reserved class '**.finfInferred**' to be used as a default class in CSS making it possible to distinguish inferred triples from asserted triples using style. In our proposed approach we used a recognizable style presentation for inferred triples by demonstrating a Semantic Web user interface representation using a preserved class for `fresnel:valueStyle` named '**finfInferred**'. This provided a style in our referred CSS file for elements containing the class-attribute '**finfInferred**'. Our results show that it is possible to define default displaying formatting information for a Semantic Web user interface definition for inferred data using Fresnel and CSS.

4.3. Cascading inferencing styling options

In this paragraph we present an option to alter the presented displaying style for specific inferred data. We setup a priority chain with our 'default' class at the lowest point. First, we gather the information from our ontology in Protégé.

4.3.1. Cascading inferencing style in Protégé

The style for inferred triples is presented as a yellow background and dotted border in the user interface of Protégé. In the ontology the information on the relation between the individuals named 'Movement_X and Car_A' is predefined. Additional information is provided at the inferred triple at the right-hand side by pressing the question-mark button. In the pop-up screen of the question-mark button, human readable information about the relation can be found. We found that our example the predicate 'is_assigned_to' is defined as '`owl:inverseOf`' the predicate 'has_Rental_Car'. The displaying information is always set to the yellow background and dotted border.

We could not find any options or settings in the menus of Protégé to alter the interface presentation of the notification of specific inferred triples formatting in the user interface of Protégé. The fact that 'Car_A is_assigned_to Movement_X' is an inferred inverse relation of 'Movement_X has_rental_car Car_A' is not styled differently in Protégé than other inferred triples. Examples of other specific inferred triples can be found in the work done by Bouwer (Bouwer, 2019).

4.3.2. Cascading inferencing style in in Fresnel

In a similar approach as in the previous paragraph we propose to extend our inferred style for inverse inferred triples by adding a reserved class style for inverse inferred triples to the default displaying style information for the Semantic Web interface. We use the defined predicate for the triple 'Car_A is_assigned_to Movement_X' where the relation is defined as '`owl:inverseOf`'. We propose to use `owl:inverseOf` for the selection of inverse inferred triples using our '**`finf:InverseInferredSelector`**' as a subclass of our previously defined **`fresnel:valueSelector`**. We define as our reserved class '**`finf:InverseInferred`**'. A class style for **`finf:InverseInferred`** that cascades over our previous defined inferred class '**`finf:Inferred`**' is connected to a class style using `Fresnel:use` **`finf:InverseInferredFormat`**. Code parts are shown in Example 6. The style specifications we use are created based on existing Fresnel definitions (both core and extended), and we propose additions to the Fresnel extended vocabulary (W3C, 2005).

```
finf:InverseInferredSelector     owl:inverseOf       rdf:type ;
                       rdfs:subClassOf fresnel:valueSelector ;
                       fresnel:propertyFormatDomain finf:Inferred ;
                       rdfs:domain   finf:Inferred..

finf:InverseInferred       rdf:type      rdfs:Class ;
                       fresnel:group:finfGroup ;
                       fresnel:use   finf:InverseInferredFormat .
```

Example 6: finf:InverseInferred selector:

Our **`finf:InferredLens`** purpose in Fresnel is intended to indicate a lens for our inferred triples by using a class. For Semantic Web browsers, it might be profitable to show a lens with purpose `fresnel:defaultLens` in situations where there are several applicable lenses (Pietriga et al., 2006). We propose to add a new lens named **`finf:InverseInferredLens`** to the Fresnel extended vocabulary for inverse inferred triples. Our proposed lens **`finf:InverseInferredLens`** to select inverse inferencing is defined as shown below in Example 7 for our scenario.

```
finf:InverseInferredLens       rdfs:subclassOf         fresnel:Lens .

finf:InverseInferredSelector   owl:inverseOf           rdf:type ;
                               rdfs:subclassOf         fresnel:valueSelector;
                               rdfs:domain             finf:Inferred;

fresnel:instanceLensDomain "SELECT ?value WHERE (?value RentalCar:is_assigned_to
?Car_Movement . ?value owl:inverseOf )" ^^fresnel:sparqlSelector.
```

Example 7: Inverse inferred sublens

The lens **`finf:InferredLens`** defines a default view on **`finf:Inferred`** containing all properties. A Fresnel browser would start with displaying **`finf:InferredLens`** because we set the purpose as `fresnel:defaultLens`. Our sublens **`finf:InverseInferredLens`** displays an inverse inferred triple. The `fresnel:sublens` property specifies that the **`finf:InverseInferredLens`** should be used to display known inverse inferred triples. We use a reference for our reserved class style for inverse inferred triples named '.InverseInferred' to our finf.css for our display styling aspects shown in Example 8.

```
:InverseInferredFormat
    a                               fresnel:Format ;
    fresnel:group                   :finfGroup ;
    fresnel:propertyFormatDomain    finf:InverseInferred ;
    fresnel:valueStyle              "InverseInferred"^^fresnel:styleClass .
```

Example 8: Fresnel format for InverseInferred

We extend the CSS for our Inferred triple by adding style based upon the type of inverse inferred triple.

```
.InverseInferred {
            border-style: solid;
            background-color: lightblue;
        }
```

Figure 11: Reserved Class '.InverseInferred' in CSS code

We create a reserved class named 'InverseInferred'. The referred reserved class '.InverseInferred' used in our finf.css applies a solid border style and light blue background color as shown in Figure 11. Our applied finf.css can be altered within the GUI of SMW. Our lens is not used when the inverse inferred triple information is absent. The default lens is used and with that the corresponding class in the CSS is used for displaying information.

4.3.3. Cascading inferencing style in SMW

We propose to add our reserved CSS class '.InverseInferred' to the CSS the browser uses in the active Semantic Web browser interface to make inverse inferred data distinctive for end users. We add our reserved CSS class named '.InverseInferred' to our connected finf.css to apply our defined style for our fresnel:valueStyle 'InverseInferred', as shown in Example 9 in red. The MediaWiki XML code part for our SMW presentation is presented as example code in Example 9.

```
<div class='InverseInferred'>
 {{#arraymap:{{{finf-Car_A-has_destination_Branch: |}}}|,
             |xxx|[[finf-Car_A-is_assigned_to::xxx]]}}
</div>
```

Example 9: Class attribute named InverseInferred (red)

We use the research done by Van Gysel (Van Gysel, 2017) that original URIs of imported RDF entities can be used by the SPARQL endpoint as a basis for our assumption that it is possible to use our proposed data for displaying information in SMW. We show our inverse inferred triple in the SMW browser interface with our proposed styling using our reserved class value '.InverseInferred'.

By presenting a default reserved class for inverse inferred data in Semantic Web browser interface displaying style and using style for inverse inferred data presentation as shown in the mockup in Figure 12, the end user is provided with human understandable information to distinguish the inverse inferred data from other inferred data presented in the SMW default browser interface. The SMW browser interface presents the data encountered provided that the value style is defined as 'InverseInferred' in connected CSS.

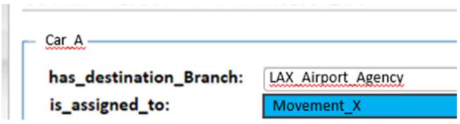

Figure 12: Mockup inverse inferred triple style to 'Car_A-is_assigned_to' with reserved class 'InverseInferred' in SMW browser interface

The provided style definitions can be altered to apply different style for presentation in Sematic Browser environment. For our reserved class '**`.InverseInferred`**' the necessary CSS code part is:

```
.InverseInferred {
            border-style: solid !important;
            background-color: lightblue !important;
        }
```

Figure 13: Code part for our reserved class .InverseInferred in CSS code

4.3.4. Conclusion

We introduced the property **`find:valueFormatType`** for `fresnel:Format` resources to make it possible to identify inferencing. We replicated the presentation of inferred data as presented in Protégé in a semantic browser environment. With the export of data from Protégé to SMW by using FForms and Fresnel, we determined the options to alter characteristics of an RDF source for how these are shown. We specified class attributes, specifically for our inferred and inverse inferred definitions. Different styles can be applied by adding our reserved classes '**`.finfInferred`**' and '**`.InverseInferred`**' to extend CSS.

4.4. Information presentation using a Details Box

Our goal in this step is to define an area containing information regarding the displayed inferred data to communicate to the user. Providing details about inferred triples guides users in understanding the given inference. Our intent is on the identification of inference information and on how to style the information of an explanation to end-users as an additional context to develop more natural language like explanations of inference. For this part of our research, we will re-use the applied design for determining the inference of our CarMovement-finf ontology from a pre-requisite to establish whether 'Car_A' 'is_assigned_to' 'Movement_X'. The rules for pre- and postcondition validation as well as consistency checking are disregarded as they do not form a pre-requisite for the scope of our user interface displaying work.

Rutledge (Rutledge et al., 2016) has demonstrated that FForms can be used to design info boxes. For our research we named 'Details Box' as an analogue for the inferred data information provided in the popup window of the Protégé user interface. We aim to extend the use of Fresnel that can be configured in SMW via the Protégé plug-in FForms to display our semantic data from inferred data within SMW.

4.4.1. Displaying information for explanation of inferred data in Protégé

Our focus in this demonstration is on the presentation of additional information with style for inferred triples we named Details Box. Our goal is to display a message on the field of the inferred triple in SMW, so the user can directly be informed of the inference. First, we focus on the information provided by Protégé.

We have shown at the beginning of this Section that Protégé notifies the end user that there can be information derived from the part of the interface providing the end user with information on the inferred data. Besides that, Protégé shows this directly to the end user by presenting a yellow background and dotted line on the interface Protégé also provides information behind a question mark button as shown in Figure 14.

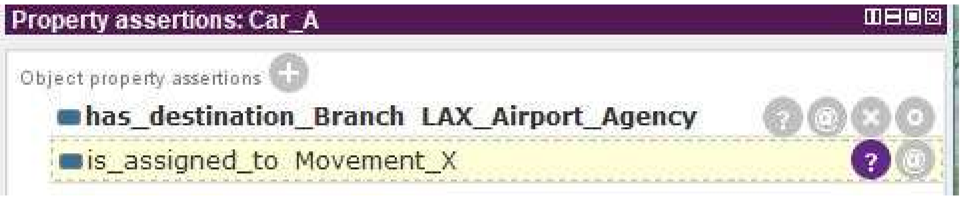

Figure 14: Protégé informs the end user of explanation of inferred data by presenting a question mark button in the user interface

When the end user moves the cursor over the question mark button at the right side of 'is_assigned_to Movement_X' the question mark button is highlighted. When the end user clicks the question mark button, Protégé will display a new window to the end user as shown at the left side and an enlarged view of the explanation itself as presented in the area below 'Explanation 1' on the right side in Figure 15.

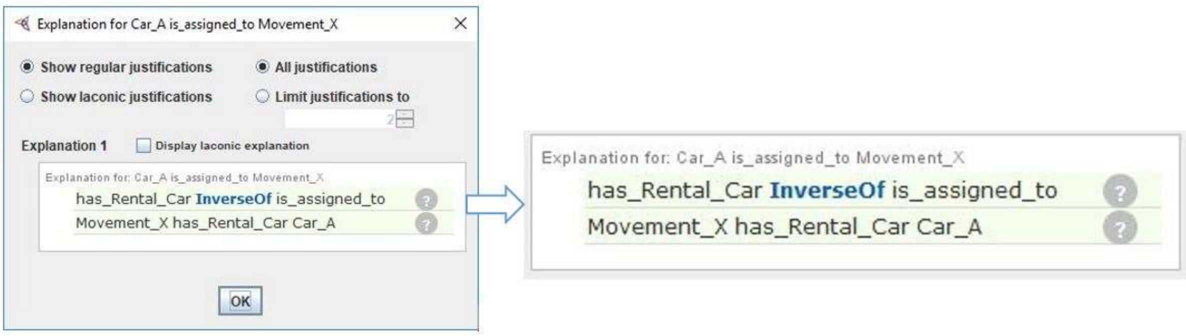

Figure 15: Popup window in Protégé user interface for presentation of extra information

Protégé shows first a label 'Explanation for:' followed by the triple 'Car_A is_assigned_to Movement_X'. Below this line Protégé presents a yellow background for the displaying explanation inferred relation 'has_Rental_Car' is the 'InverseOf' 'is_assigned_to' it has discovered to the end user as text with bold font and blue color. The last line contains the triple the inferred information is from. We could not find any options or settings in the menus of Protégé to alter the interface presentation of extra information for inferred data formatting in the user interface of Protégé.

4.4.2. Displaying information for explanation of inferred data in Fresnel

We aim to repeat the information presentation provided in Protégé in Fresnel by using a message concerning the inferencing to the end user. With the use of pseudo class selectors we aim to create selectable fields showing the message to the end user when the end user hovers over the information.

These pseudo class selectors, such as `fresnel:active` or `fresnel:hover`, are used to specify the 'behavior' of the format, such as 'hover behavior'. The 'hover behavior' can be described as changing displaying information when the end user moves an appropriate pointing device over a designated element but does not activate it (W3C, 2005). Using the Fresnel vocabulary, the required code for 'hover behavior' is defined as in Example 10.

```
fresnel:valueStyle    [ fresnel:hover    ":hover" ^^fresnel:styleClass;]
```

Example 10: Proposed defined hover behavior

We add the styling instructions for our pseudo class selectors to our finf.css for the Semantic Web displaying information representation as presented in Figure 16.

```
:hover {
        background-color: blue;
        color: white;
    }
```

Figure 16: CSS styles for pseudo-class :hover

Continuing on the results earlier in this Section we combine the output we need for our displaying inferencing information in Fresnel for displaying information to the end user in Semantic Web interfacing with the work done by Bos (J. Bos, 2018) on lens and format definitions for the violation messages and most recently the work done by Bouwer (Bouwer, 2019) on interface styles for business rule violations. Their work shows how to create (sub)lenses in Fresnel containing message information concerning the inferencing that can be presented as displaying information to the end user in the SMW environment. We propose an extension that is active when an inferred triple is present it shows the following text 'Message: **`Inferred triple`**'. When an inverse inferred triple is present the message is extended with the following text 'inverseOf' making the complete text '**`Message: inferred triple inverseOf`**'.

We define a format lens **`finf:DetailsBoxLens`** as show in Example 11 in Fresnel code.

```
finf:DetailsBoxLens          rdf:type      fresnel:Lens  ;
fresnel:classLensDomain      finf:Inferred        ;
fresnel:use                  finf:DetailsBoxFormat .


finf:DetailsBoxFormat       rdf:type       fresnel:Format  ;
fresnel:valueFormatDomain   finf:inferred ;
fresnel:resourceFormat      [ fresnel:contentBefore
                             "Message: inferred triple "^^xsd:string ] ;
fresnel:valueFormatDomain   finf:InverseInferred ;
fresnel:resourceFormat      [ fresnel:contentAfter
                             "inverseOf "^^xsd:string ] .


finf:DetailsBoxFormat        rdf:type      fresnel:Format  ;
fresnel:propertyFormatDomain finf:inferred ;
fresnel:valueStyle            ":hover"^^fresnel:styleClass  .
```

Example 11: Sublens named finf:detailsBox

For our model, we declare **`finf:DetailsBoxLens`** which applies to instances of class **`finf:Inferred`** as specified by property `fresnel:classLensDomain`. Our Fresnel implementation is intended to display additional information when hovered over using a `Fresnel:valueStyle ":hover"` as a `fresnel:styleClass`. The styling instructions for our pseudo-class ':hover' are stated in our stylesheet finf.css. Our **`finf:DetailsBoxLens`** is to be used to display details about inferred triples. We add meaning and message to be able to add details concerning the inferred triple to be displayed in SMW interface.

4.4.3. Mockup displaying information for details of inferred data in Semantic MediaWiki

Our proposed data properties defined in this Section and our proposed Fresnel purpose individual addition to the extended vocabulary are added to the CarMovement-finf ontology. The Details Box itself applies as a selection mechanism on mouse over for the inferencing information we aim to show with our model. The FForms we use partially to determine how the inferencing properties should be displayed (Theunissen, 2015). This way we use the flexibility of possible implementation to achieve our goal for our Details Box in SMW by using Semantic Web technologies in our model. To show the Details Box on mouse over by using pseudo class `':hover'`, the element containing the pseudo class can be referred to in a general CSS (W3C, 2005). A system administrator could deliver a SMW user interface using Fresnel user interface definitions and selector queries we proposed in our model. The delivered displaying information for the end user could resemble the mock-up shown in Figure 17.

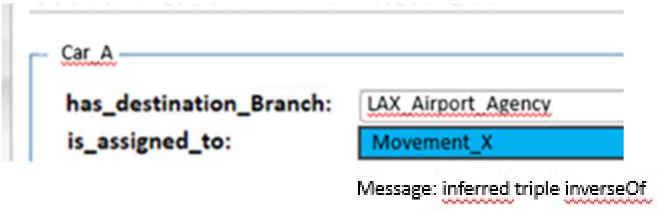

Figure 17: Mockup of 'hover' behavior when moved over in SMW

The details displayed below the inferred triple could be retrieved from the `message` and annotation properties to our **`finf:inferredLens`** and **`finf:inverseInferredLens`** of the rule class submitted as parameters. CSS can, for example, add text content, although it would be the same text every time. That (universal) text would be something like "**`Message: inferred triple inverseOf`**". The DetailBoxLens is intended to use for displaying details concerning the inferred triple by displaying the message and the meaning. Using a lens provides further styling options as shown previously in this section. FForms cannot be used in its current form to generate displaying information of inferred data for a Semantic Web user interface. However, if an altered FForms is used it should be feasible to generate the required displaying information.

4.4.4. Conclusion

Although our work shows it is possible to use Semantic Web technologies how the displaying of information to the end user could be presented and suggestions on how to implement the displaying information for the use in a Semantic Browser user interface. Due to FForms limitations to implement inferred data displaying information for Semantic Browser user interface upon the CarMovement-finf ontology we suggest further research to extend the use of FForms. We propose to include inferred and inverse properties in FForms to be able to use the inferred and inverse information. This makes it possible to provide each direction with its own `rdfs:label`. In addition, we suggested that the inverse properties could be added to be able to add such labels for incoming links as a style-like activity. Furthermore, our mockups of displaying information for the Semantic Browser user interface examples support Fresnel definitions.

Our work on displaying of information enables a software engineer to develop a displaying of information in a Semantic Web user interface presenting inferred data and related information. As a next step, it also seems possible to generate displaying information for other Semantic Web browser user interfaces other than SMW environment because of the generic setup of our ontology. The feasibility cannot be determined with the help of our prototype work.

5. Discussion, conclusions, recommendations and reflection

5.1. Discussion

We create and define ontologies for inferencing, and we made Semantic Web code for an example population. We create a mockup implementation for the inferencing ontology to define our rules for displaying style in Semantic Web browser, based on how Protégé presents style components for inferencing in the user interface. Another possible option for presenting style in Semantic Web user interface for inferencing is for example by embedding the style presentation in a controller of Semantic Web user interface. In Protégé, we did not discover a way to alter the provided displayed styling in the user interface. The end user is not supported in Protégé if alterations to the displayed styling for inferencing are beneficial to the end user. We propose for the end user a controller with options to select alternative display styling next a default Semantic Web user interface configuration from a selection of predefined style components.

5.2. Conclusions

The main question for our research which we want to answer in this thesis is:

"How and to what extent can one specify interface style for data that is inferred on the Semantic Web?"

We propose an extension to FForms to present detailed information to the end user and providing possible options in order to export inferencing information to Fresnel. We present **`finf:Inferred`** and **`finf:inverseInferred`** as new individuals within the fresnel:Group class to a displaying inferred data user interface lens. We propose **`finf:inferredLens`** and **`finf:inverseInferredLens`** to be distinguished from other lens purposes. To determine if a triple is inferred, we introduced **`finf:infSelector`** as a new subclass of Fresnel to be able to select the inferred value.

The results show that it is possible to define style displaying formatting information for a SW user interface definition for inferred data using Fresnel. We are able to replicate the presentation of inferred data as presented in Protégé in a SMW browser environment to some extent. With the export of data from Protégé to Fresnel by using FForms and (sub)lens specifications, we determine the options we have to alter characteristics of an RDF source for how these are shown.

In answering our main question, we conclude that we can determine how the selected properties are generated. We specify for 'formatting attributes', specifically for our CarMovement-finf. We provide a link to our CSS, with which the class styles and pseudo-class styles are determined, for the presentation of the inferred data on Semantic Web browser user interface. We propose default styling in the linked CSS can be overridden by an end user by applying a preferred custom style as an override to our preferred styling options in the linked CSS. Such source files for this thesis are online at http://is.cs.ou.nl/OWF/index.php5/Masters_Thesis_Pascal_Mellema.

5.3. Practice recommendations

We use Semantic Web technologies Fresnel, CSS and Semantic Web browser to support our implementation method for displaying inferencing style. The options provided by the used languages for implementing our displaying inferencing styles is supported by specific enhancements to support the displaying of inferred data as a separate type of information.

As described as part of our conclusions, a practical follow up that could be taken on basis of this work is the development of an extension to Fresnel and including default CSS additions for inferred data style to SW interfaces. This should result in Semantic Web supported user interface style components to directly deliver a Semantic Web user interface with displaying information for inferred data.

5.4. Recommendations for further research

We propose and partially demonstrate an extension of Fresnel which implements rule styles for inferred data. Further extensions to implement other rule styles are necessary to get a comprehensive approach for dealing with rule styles. Violation rule styles are covered in the research of Bouwer (Bouwer, 2019). Another addition would be to have styling which supports the user to distinguish different types of values of other fields. In our model we suggest additions to Fresnel and used other supported Semantic Web technologies. We presume our assumptions need to be fully evaluated before this work can be published.

5.5. Reflection

With our work we have built our model to explain how aspects of the Protégé user interface can be used as a target for creating an analogue of displaying information in a Semantic Web browser environment. Our model contained ideas and concepts, and included a mechanism to derive information and add style information for inferred data.

We demonstrate with our work it is possible to use Semantic Web technologies to present displaying information to the end user in a human readable way. We show how displaying information could be presented by using style and provided suggestions on how to implement the displaying information for the use in a Semantic Browser user interface. Our mockups of displaying information for the Semantic Browser user interface examples supports Fresnel definitions. Our work on displaying of information enables a software engineer to develop a displaying of information in a Semantic Web user interface presenting inferred data and related information.

References

Allemang, D. H., J. (2008). *Semantic Web for the Working Ontologist - Modeling in RDF, RDFS and OWL*. Amsterdam, The Netherlands: Morgan Kaufman Publishers (Elsevier).

Bos, J. (2018). *Specification of a User Interface Template for Semantic Web-implemented Hohfeldian Right and Duty Legal Rules using the Fresnel Data Interface Language.* (MSc), Open University of the Netherlands, Heerlen.

Bos, P. (2013). *Bedrijfsregels in verschillende vormen - Een vergelijking op toepasbaarheid tussen SWRL en Relatie algebra bij wetteksten.* (Dissertation/Thesis), Open Universiteit Nederland, Retrieved from http://hdl.handle.net/1820/4977

Bouwer, E. (2019). *Patterns of and extensions to semantic browser style specification for business rule violations.* (MSc Work in progress), Open Universiteit Nederland, Heerlen.

Business Rules Group. (2000). *Defining Business Rules - What Are They Really?* Retrieved from http://www.businessrulesgroup.org: http://www.businessrulesgroup.org/first_paper/BRG-whatisBR_3ed.pdf

Coenen, A. H., L; Van Roosmalen, M; Spreeuwenberg, S. (2008). *Uw bedrijf geregeld met business rule management*. Den Haag: Sdu Uitgevers.

Frias, L., Queralt Calafat, A., & Olivé Ramon, A. (2003). EU-Rent car rentals specification.

GEIST Research Group. (2015b, 19-09-2017). SBVRwiki - business rules in wiki with SBVR. Retrieved from http://loki.ia.agh.edu.pl/wiki/eu-rent:start

Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems, 19*(2), 87--92.

Joosten, H. (2014). AmpersandTarski/ampersand-models/EU-Rent. Retrieved from https://github.com/AmpersandTarski/ampersand-models/tree/master/EURent

Joosten, S., Wedemeijer, L., & Michels, G. (2010). *Rule Based Design* (December 2010 ed.). Heerlen: Open University of the Netherlands.

Kuba, M. (2012, 2012). OWL 2 and SWRL Tutorial. Retrieved from https://dior.ics.muni.cz/~makub/owl/swrl_tutorial_rule3_inference.png

MediaWiki.org. (2018, 2018, August 26). Extension:Page Forms. Retrieved from https://www.mediawiki.org/w/index.php?title=Extension:Page_Forms&oldid=2863609

Object Management Group. (2008). Semantics of Business Vocabulary and Business Rules (SBVR), v1.0. In.

Pietriga, E., Bizer, C., & Karger, D. (2006). Fresnel: A Browser-Independent Presentation Vocabulary for RDF. In (Vol. 4273, pp. 158-171). Berlin, Heidelberg: Springer Berlin Heidelberg.

Reynares, E., Caliusco, L., & Galli, R. (2014). SBVR to OWL 2 Mappings: An Automatable and Structural-Rooted Approach. *CLEI ELECTRONIC JOURNAL, 17*(3), Paper 2.

Rutledge, L. (2013). *From Ontology to Wiki – Generating Cascadable Default Fresnel Style from Given Ontologies for Creating Semantic Wiki Interfaces*. Paper presented at the 10th Extended Semantic Web Conference (ESWC 2013), Monpellier, France.

Rutledge, L., Brenninkmeijer, T., Zwanenberg, T., van de Heijning, J., Mekkering, A., Theunissen, J. N., & Bos, R. (2016). From Ontology to Semantic Wiki – Designing Annotation and Browse Interfaces for Given Ontologies. *Semantic Web Collaborative Spaces, 9507*, 53-72.

Saunders, M., Lewis, P., & Thornhill, A. (2007). *Research Methods for Business Students*: Financial Times/Prentice Hall.

Slootweg, P. R., L.; Wedemeijer, L.; Joosten, S. (2016). The Implementation of Hohfeldian Legal Concepts with Semantic Web Technologies. *AI4J – Artificial Intelligence for Justice*, 8.

Spreeuwenberg, S., & Gerrits, R. (2006). *Business Rules in the Semantic Web, Are There Any or Are They Different?* (Vol. 4126).

Stanford Center for Biomedical Informatics Research. (2016). Protégé. Retrieved from https://protegewiki.stanford.edu/wiki/Main_Page

Theunissen, T. V. d. H., Joop ; Mekkering, Alex. (2015). *Afstudeerscriptie ABI - Team 30.* (Bachelor T61327 - Afstudeerproject bachelor informatica), Open Universiteit, Heerlen.

Van Gysel, H. A. J. (2017). *Implement Agile Development Process with Business Rules on the Semantic Web.* (Master Software Engineering), Open University of the Netherlands, Retrieved from http://dspace.ou.nl/

W3C. (2005, 20050726). Fresnel - Display Vocabulary for RDF. Retrieved from https://www.w3.org/2005/04/fresnel-info/manual/

W3C. (2019). Cascading Style Sheets. Retrieved from https://www.w3.org/Style/CSS/

W3C. (2019). Cascading Style Sheets. Retrieved from https://www.w3.org/Style/CSS/

Appendix A

Overview of finf ontology code.

```
@prefix :
<https://is.cs.ou.nl/OWF/index.php5/Masters_Thesis_Pascal_Mellema/finf#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix finf:
<https://is.cs.ou.nl/OWF/index.php5/Masters_Thesis_Pascal_Mellema/finf#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix EURent:
<http://desibo.frsf.utn.edu.ar/ontologies/2012/9/CarMovement.owl#> .
@prefix fresnel: <http://www.w3.org/2004/09/fresnel#> .
@base <https://is.cs.ou.nl/OWF/index.php5/Masters_Thesis_Pascal_Mellema/finf> .

<https://is.cs.ou.nl/OWF/index.php5/Masters_Thesis_Pascal_Mellema/finf> rdf:type
owl:Ontology .


fresnel:stylesheetLink     rdf:type      owl:AnnotationProperty .

:finfGroup           rdf:type      owl:NamedIndividual ,
                     fresnel:Group ;
                         fresnel:stylesheetLink
<http://is.cs.ou.nl/OWF/index.php5/Masters_Thesis_Pascal_Mellema/finf.css> .


finf:Car_A           rdf:type      owl:NamedIndividual .

finf:Movement_X            rdf:type      owl:NamedIndividual ;
                     EURent:has_Rental_Car      :Car_A .



finf:infSelector     rdfs:subClassOf finf:Inferred ;
                     fresnel:propertyFormatDomain finf:Inferred ;
                     rdfs:domain   finf:Inferred ;
                     fresnel:instanceLensDomain
                      "SELECT ?value WHERE
                      (?value RentalCar:is_assigned_to ?Car_Movement .
                       ?value rdf:type finf:Inferred)" ^^fresnel:sparqlSelector .


finf:Inferred        rdf:type      rdfs:Class ;
                     fresnel:group:finfGroup ;
                     fresnel:use   finf:InferredFormat .

finf:InferredFormat rdf:type      fresnel:Format ;
                     fresnel:propertyFormatDomain finf:Inferred ;
                     fresnel:valueStyle "finfInferred"^^fresnel:styleClass .


EURent:has_Rental_Car    rdf:type owl:ObjectProperty .


finf:InverseInferredSelector     owl:inverseOfrdf:type ;
                     rdfs:subClassOf fresnel:valueSelector ;
                     fresnel:propertyFormatDomain finf:Inferred ;
                     rdfs:domain   finf:Inferred ;
                     fresnel:instanceLensDomain
                      "SELECT ?value WHERE
                      (?value RentalCar:is_assigned_to ?Car_Movement .
                       ?value owl:inverseOf )" ^^fresnel:sparqlSelector .
```

```
finf:InverseInferred      rdf:type     rdfs:Class ;
                    fresnel:group:finfGroup ;
                    fresnel:use  finf:InverseInferredFormat .

finf:InverseInferredFormat rdf:type     fresnel:Format ;
                    fresnel:propertyFormatDomain finf:Inferred ;
                    fresnel:valueStyle "InverseInferred"^^fresnel:styleClass .

finf:DetailsBox     rdfs:subClassOf     finf:Inferred ;
                    owl:intersectionOf (EURent:Car_Movement
                   [ owl:onProperty EURent:is_assigned_to ;
                    owl:hasValue "true"^^xsd:Boolean]);
                    finf:message "Inferred triple" .
```